AMENDMENTS TO THE SPECIFICATION

Please replace the following paragraphs in the Specification as follows:

On page 35, please replace the paragraph beginning on line 19 with the following:

-- Fig. 18 starts at the beginning of the card table 320. If the card table has been fully processed 322, the system has completed a pass summarizing potentially modified locations. But if the card table has not yet been completely processed 324, the next eight bytes are loaded 326 in a location V for processing. The bytes are checked for zeros or SCAN values 328 indicating dirtied memory cards. If any of the checked bytes indicate the presence of dirtied memory cards, a copy of the eight bytes is made that is identical to the previously read eight bytes except any that are zero (indicating a dirty card) are changed to be the SCAN value. A CAS is performed to change the eight entries in the card table from the previously read values to the new values. If successful, the new entries have replaced the old ones in the card table 331 and the entries with SCAN values may now be scanned. Having succeeded the new sequence including SCAN values becomes the current value of V'. If the CAS fails 333, additional entries in the eight-byte sequence of the card table must have been marked as dirty by the application, and so, to avoid summarizing references the application is currently modifying, control is returned to inquiring if the card table is exhausted. The corresponding dirtied memory cards are scanned 330, remembered sets are updated, and references from younger generations are summarized. When the dirtied memory cards have been processed the contents of the word V' are updated into location V". The locations of V' and V" are processed with a CAS instruction 332 and control returns to see if the card table has been completed or exhausted. It does not matter if the CAS succeeds or fails, because any dirtied locations that occurred while the memory cards were being processed are retained in the card table for a future collection. If the CAS succeeds then all of the memory cards whose card table entries had been marked dirty prior to the scanning of the cards have had their entries updated to reflect the results of the scanning. In either case the collector continues on as before. Fig. 19 is identical to the operations described for Fig. 18 up to the point of forming a V" 330. A CAS is performed 332 from location V' to V" (in this terminology V' is the old value and V" is the new value) where, if the CAS succeeds 334, then no modifications were made during the processing of the card table and updating references. However, if the CAS fails 336, since modifications were made

concurrently, then for each newly dirtied byte in the V' the corresponding byte in V" is dirtied by placing a zero in that byte. In this manner the newly dirtied bytes are recorded while maintaining the just-performed summarization for card table entries that have not just been dirtied; the newly dirtied entries will eventually be processed during a later collection interval. --

Please replace the paragraph on page 37, beginning on line 7, with the following:

-- Other atomic operations may be used above instead of the CAS operation. Some modern processors, for example, provide a pair of instructions, load-locked (LL) and store-conditional (SC) that serve the same purpose as, and instead of, the CAS operation. On these processors, LL/SC are suitable for implementing the invention. Similarly, more limited atomic instructions (such as the SPARCV9 architecture's load-store-unsigned-byte (LDSTUB) instruction that atomically reads a byte and changes its contents to all ones) may be used for selectively setting indicators to empty if the empty state is represented by all ones. --